CSCE 230 – Lab 3: Assembly Array Programs

Due: 11:59PM, Sep 12 (Tuesday)

Objectives

- Learn the basic syntax and structure in Nios II assembly programming
- Learn how to use branching and shift instructions
- Learn how to use shift and bitwise logic instructions
- Learn how to use load and store instructions
- Learn how to use arrays

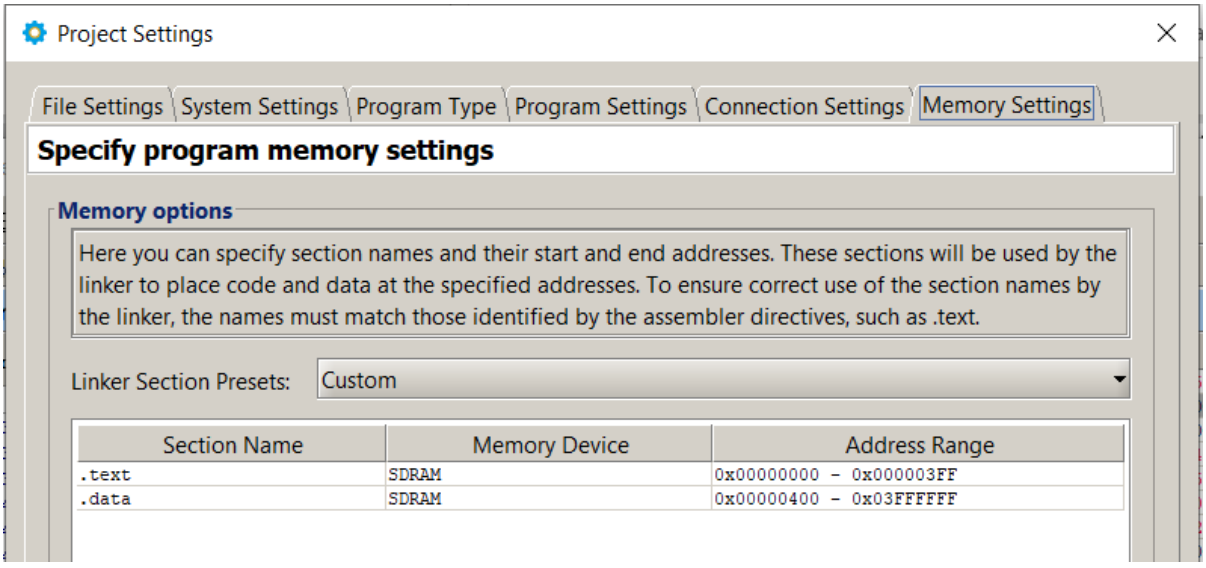
Useful References on Canvas

- Lecture notes for Chapter 2 and Appendix B
- Altera Nios II Processor Document
- Altera Nios II Instruction Document

Lab: Assembly Array Program

First, please create a new project

- Create a new project directory. If you are using the lab computers, it is more convenient to create a new project directory in C:\temp (after the lab, please backup your files to your Z: drive)
- In your project directory, create an empty text file using any text editor, and name it lab3.s or any other names, but the file extension should be .s
- Open the Altera Monitor program
    - Menu "File" → "New Project"
    - In the "New Project Wizard",
        - Select your project directory and specify a project name, and then click "Next".
        - Select system "DE10 Light Computer", and then click "Next"
        - Select program type "Assembly Program", and then click "Next"
        - Add your lab3.s file into the project, and then click "Next"
        - Use the default system parameters, and then click "Next"
        - For the memory settings, use the default SDRAM memory device with custom memory ranges as shown below, **set .text to 0 and set .data to 0x400** (you do not need to type 0x) **for this lab**, and then click "Finish"
        - **Note: The .text and .data memory sections cannot overlap!**



Then, open your lab3.s file to write a single assembly program with the following data section to complete the following three tasks using the code on the following page.

Three Tasks

- Task 1: Please find the **maximum** element in the first array.
  - The array is located at address *array1*.
  - The array has a total of 4 elements.
  - Each element is an 8-bit unsigned binary number
  - Please find the maximum element, and write it (8 bits) to the memory at address *array1_max*.
- Task 2: Please find the **minimum** element in the second array.
  - The array is located at address *array2*.
  - The array has a total of 4 elements.
  - Each element is a 16-bit unsigned binary number
  - Please find the minimum element, and write it (16 bits) to the memory at address *array2_min*.
- Task 3: Please count the **total number of bit 1's** in all the elements of the third array.
  - The array is located at address *array3*.
  - The array has a total of 3 elements.
  - Each element is a 32-bit binary number
  - For each element (32 bits), please count the total number of bit 1's in the element. For example, there are a total of 8 bit 1's in the first element 0x00112233 = 00000000 00010001 00100010 00110011
  - Please count the total number of bit 1's in all the elements, and write it (32 bits) to the memory at address *array3_bitone*.
  - Hint: You may use the shift or rotation instructions and bitwise logic instructions to check whether each bit of a binary number is 1 or 0.

```
/*-------------------------------------------------------------------
Name:  <Your Name>
Date:  <The date you stated working on the file>
Course:      CSCE 230 - Computer Organization
File:  <This file's name>
HW:    Lab 3 - Assembly Array Programs

Purp:  This program is a demonstration in the basic syntax and
       structure in Nios II assembly programming by using arrays.

Doc:   <list the names of the people who you helped>
       <list the names of the people who assisted you>

Academic Integrity Statement: I certify that, while others may have
assisted me in brain storming, debugging and validating this program,
the program itself is my own work. I understand that submitting code
which is the work of other individuals is a violation of the honor
code.  I also understand that if I knowingly give my original work to
another individual is also a violation of the honor code.
-------------------------------------------------------------------*/
       .text               # located at address 0
       .global      _start
_start:
       ....
       ....
       Your code here
       ....
       ....
end:br      end

      .data         # located at address 0x400

# data for the first array
array1:
       .byte  0x00, 0x11, 0x22, 0x33
array1_max:
       .byte 0

       .skip 3

# data for the second array
array2:
       .hword 0x0123, 0x4567, 0x89AB, 0xCDEF
array2_min:
       .hword 0

       .skip 2


# data for the third array
array3:
       .word  0x00112233, 0x44556677, 0x8899AABB
array3_bitone:
       .word 0

       .end
```

CSCE 230 – Lab 3: Assembly Array Programs

Programming requirements

- You must write only a single assembly program
- You must use the data section defined in the previous page, and the .data must be set to 0x400 in your Monitor program (described on the first page).
- Your code must have no more than 256 machine instructions; otherwise, it will overlap with the data section.

Quick correctness check:

- After running your program,
  - the byte at address array1_max should be 0x33.
  - the halfword at address array2_min should be 0x0123
  - the word at address array3_bitone should be 0x28

Please answer the questions on Canvas.

- To check the words in your data section, you may go to the "Memory" tab, type "400" into the "Address (hex)" box, and click the "Go" button.